Note: This is longer than the actual midterm will be. It covers only a sampling of topics that might be on the exam.

1. (a) Prove that if $f(n)$ and $g(n)$ are polynomials with the same degree, then $f(n) \in \Theta(g(n))$.
   (b) Prove that for every real number $k$, $n^k \in o(2^n)$.

2. (a) In this question only you may use any theorems proved in class, but say how they are being used and show that any hypotheses to the theorem are true. Find a $\Theta$ estimate for $T(n)$ if:

   $$T(n) = 8T(\lceil n/3 \rceil) + n^2 \log(n)$$

   (b) Explain why the Master Theorem for divide and conquer recurrences cannot be used to solve the recurrence

   $$S(n) = S(n/2) + \log(n)$$

   (c) Explicitly (no big-$O$) solve the recurrence in part (b) when $n$ is a power of 2.

3. Consider the following problem:

   **Instance:** A list $X = x_1, \ldots, x_n$ of integers
   
   **Output:** The index of the largest integer in $X$.

   (a) Give a pseudocode description of an efficient iterative (nonrecursive) algorithm that solves this problem.
   (b) Give a formal correctness statement for your algorithm and prove the algorithm is correct using the method of invariants.

4. Give a high level description of a data structure for maintaining a set of integers with the usual operations $INSERT$, $DELETE$, and the new operation

   $STAT(k$: integer): return the integer with rank $k$ (the $k$th largest integer).

   You may describe your data structure by saying how to modify data structures we have discussed in class, and just say briefly in words how $INSERT$ and $DELETE$ must be modified. Give pseudocode for $STAT(k)$. All operations should take time $O(\log(n))$, where $n$ is the number of integers in the set.

5. (a) Give asymptotic estimates for the worst case complexity and average case complexity for basic QuickSort.
(b) State a sharp lower bound on the number of comparisons used by a comparison based sorting algorithm.

(c) Let $H$ be a min-heap with $n$ elements, stored in an array in the usual way. Give a pseudocode description of an efficient implementation of the operation $INSERT(H, x)$ that inserts item $x$ in heap $H$.

6. Give a high level description of a data structure that supports the following operations:

- $Insert(x)$: insert key $x$ in the structure only if it is not already there.
- $Delete(x)$: delete key $x$ from the structure if it is there.
- $Next(x)$: return a pointer to the smallest key in the structure that is larger than $x$ (or a NIL pointer if $x$ is the largest key in the structure).

You may refer to structures and algorithms we have described in class in describing $Insert$ and $Delete$. All operations should take time $O(\log(n))$.

7. Insert in order into a red-black tree: 7, 3, 2, 9, 11, 15. Show the tree after each insertion.

8. (a) Describe an efficient algorithm that determines whether a given undirected graph $G = (V, E)$ is a tree.

(b) Analyze the running time of your algorithm if the graph is represented by an adjacency list and if the graph is represented by an adjacency matrix.